[redacted] SECRET

ILLEGIB

ORD #4977-68

F I N A L R E P O R T

BY

COMPUTER SECURITY TECHNICAL PANEL

submitted to

DEFENSE SECURITY BOARD STEERING GROUP

FOR

TASK FORCE ON COMPUTER SECURITY

July 15 1968

May be downgraded to SECRET upon removal of appendix

NSA, OSD reviews completed

[redacted] SECRET

ILLEGI

Copy 15 of 24 copies.

TABLE OF CONTENTS

SECTION 5 SECRET

SECURE SYSTEMS

APPENDIX TOP SECRET CRYPTO

DETAILED DISCUSSION AND RECOMMENDATIONS FOR CRYPTOLOGIC RESEARCH

The following is a list of members of the Technical Panel

Edward L. Glaser - Chairman
Case Western Reserve University

Arthur A. Bushkin - Secretary
Massachusetts Institute of Technology

James P. Anderson
Anderson and Company

Edward H. Bensley
The MITRE Corporation

Charles R. Blair
Associated Universities, Inc.

Harold M. Jayne
Executive Office of the President

P.L. 86-36

[redacted] (ex officio) - Chairman, Policy Panel
National Security Agency

STAT

Lawrence G. Roberts
Advanced Research Projects Agency

Jerome H. Saltzer
Massachusetts Institute of Technology

Willis H. Ware (ex officio) - Chairman, Task Force
The RAND Corporation

Section 1. Introduction, Recommendations and Summary

1.1 The Background of the Report

In late 1967 a Task Force was established under ARPA at the behest of DDR&E. The purpose of this Task Force was to determine the problems of creating secure time sharing systems. As a part of this Task Force the technical panel was established. This panel met quite frequently during late 1967 and into 1968. This work culminated in a workshop being held at the Communications Research Division of the Institute of Defense Analysis at Princeton, New Jersey, from March 28-30, 1968. The following report is the output of this workshop.

1.2 Recommendations

The bulk of this report will be concerned with various technical facilities that can and must be included in order to make a time sharing system secure. The purpose of this set of recommendations is to indicate those research areas that must be pursued in order to guarantee this security. It should be further emphasized that no attempt has been made to delineate either the cost of the various research tasks or to indicate within this report who should be tasked with the various areas of research. Rather, the specific research programs are delineated within other sections of this report and should be self-evident as to the cognizant agency. The following are the four primary research areas that should be pursued. The first two can be considered to be short term, while the second two are long term.

1. Security structure language. The design of the security structure language should be completed and its implement algorithm defined. This package to be submitted for review and approval at the earliest possible date (see section 3).

2. Consistency checks. A rapid early analysis should be made of the possibility of incorporation hardware consistency checks in equipment supplied by major manufacturers today (see section 4).

3. Systems certification. A research program should be delineated for the problem of determining the feasibility of more automated, hence exhaustive, certification of the integrated hardware/software system with due regard to its operational environment (see sections 4 and 5).

4. Cryptologic research. A program for the necessary cryptologic research to be initiated as soon as possible in order to facilitate the early availability of secure time sharing systems (see section 5 and the Appendix).

1.3 Summary of the report

The remainder of this report is divided into four sections plus an appendix. Section 2 of this report is concerned primarily with the interaction between the technical and the procedural, doctrinal, problems. It deals with certain facilities that are expected to be present in order to make the system operate properly from the doctrinal standpoint. Section 3 concerns itself with a definition of a security structure language and is aimed primarily at a view of the system from the standpoint of the security officer. What is contained in this section can best be described as a generalization for what must be implemented in order to facilitate the present structure of the security system of the United States Government. Section 4 primarily concerns itself with state of the art techniques with regard to the implementation of secure time sharing systems. As will be pointed out in this section, true secure time sharing systems are currently beyond the state of the art, but this section does concern itself with an attempt at making privacy systems and gives guide lines to facilitate the design of such systems in the near term. Section 5 concerns itself with the longer term secure time sharing systems and discusses in some detail the two techniques that are currently known by which time sharing systems can be made to be secure. The appendix gives greater detailed discussion on the area of the cryptologic research. It was felt advisable to separate out this kind of detail from the rest of body of the report in order to keep the security level of the entire report somewhat lower than what otherwise would be necessary.

1.4 Acknowledgments

It is impossible to list all of the contributors to this report, however, it is fitting here to give particular thanks to the members of the technical panel, the guests of the panel at our various meetings and particularly at the meeting at Princeton, and further to the personnel of the Communications Research of the Institute of Defense Analysis for acting as hosts and giving us such fine accommodations during the extremely intensive three days of workshop. Finally, it would be very remiss not to give particular thanks to the various secretaries of the members of this panel who have given unstintingly of their time and services to make this report possible.

Approved For Release 2005/06/09 : CIA-RDP71R00510A000200130003-7

2. TECHNICAL/POLICY INTERACTIONS

2.1 Purpose

Although much of the material in this section is covered in other sections of this report, its treatment here is from the stand point of the procedural problems involved rather than the technical feasibility. The reader desiring a cursive view of the technical problems may glean much of the information by a parusal of this section only. However, the reader should remember this is primarily a procedural view of the technical problems and therefore is an overview. More detailed coverage is contained in the following sections of this report and the appendix.

2.2 Authentication

It is suggested that multi-access resource sharing systems have significant advantages. To achieve these advantages it is necessary to have one or more computer complexes, many remote terminals and an even larger number of individuals who have controlled access to the system, the processing which the system can accomplish, and the information stored within it. The objective of security in the system is to assure that no classified information is disclosed to any individual not formally authorized to receive it. Since it is to be expected that not all individuals (users) at any one remote terminal will require access to the same material, it is necessary to employ a means by which the user can be uniquely identified to the computer system to assure that he only gains access to the information for which he is authorized. A password system using one-time material which is supplied individually to each user will serve this purpose. This

password authenticates the user to the system and is the key which unlocks the system to him up to the predetermined access limits. An authentication must always be required at log-on and may also be required upon demand by the system while the user has control of the terminal in order to assure the system that the original user is still in control. This re-authentication procedure may be invoked on the basis of the amount of time elapsed since the last authentication and/or upon the completion of a specified number of transactions as determined by the system design.

The clearance of a remote terminal in a switched communications system will not necessarily be obvious to the central facility. This will require that a means be provided for authenticating a specific terminal to the control facility to assure that only information authorized to be handled by that terminal is actually delivered to it by the central facility. Also the user must know that the distant central facility he has reached via switched communications system is indeed the terminal authorized to receive the information he is about to transmit. If the communication links from the remote terminals to the central facility are secured by crypto-equipment utilizing unique keys for each link then identification of the key will authentically establish the identification of the terminal to the central facility and further authentication may or may not be necessary. If the remote terminal is a part of a multi-holder cryptonet, a unilateral authentication using one-time passwords is required. The can be accomplished by an appropriate challenge-reply pair being exchanged between the central facility and the remote terminal. At the remote terminal appropriate

passwords will be issued to the user by individual responsible for the security control of the terminal. The user must give unused passwords protection equivalent to the highest level information that could be accessed with them.

It will be necessary for computers to transmit information to other computers in some systems. The provision of the preceding paragraph apply for establishing the authenticity of the computers one to the other over either point-to-point or switched communications systems.

In some systems a user operating at a remote terminal which is assigned to one computer facility may want to access information and/or run a job on another computer facility. This is permissable and may be done by the user authenticating himself to his own computer by the previously described techniques. The first computer then passes to the second computer, over their authenticated communications link, the information required by the second computer to permit it to determine the degree of access permitted to the user.

2.2.2 Communications Link Protection

It has been stated elsewhere that all communications links passing classified information will be secured by appropriate cryptographic equipment. Assume that a communication link between two terminals has been established and the cryptographic equipment is in synchronization. The link may or may not also have traffic flow security depending upon its security requirements. It is recognized that in any case information will not actually be passing over the link at all times. This is true on either a half-duplex or full-duplex link. In the case where the cryptographic

equipment is stepping independently of the plain text input information, there may be a danger of spoofing.* For example, a user could log on to the system, properly accomplish all the identification and authentication procedures and then apuse. During this pause no enciphered plain text is being transmitted over the link. In this situation it could be possible to tamper with the link and introduce bogus plain text which will be deciphered by the receiving terminal and treated as authentic plain text information. The consequences of this action could range from mere harassment to extremely serious. Files could be erased, programs changed, etc., depending upon the rights and privileges established by the bona fide transmitting terminal. This threat can be circumvented by proper chice and/or application of the cryptographic equipment to the system and must be accomplished as a part of the initial system design.

In the case of a half-duplex link operating in a switched communications net where the same cryptographic key is held by all terminals and used for the transmit and receive function, there may be a threat to the system due to physical capture of one terminal. User and terminal authentication procedures discussed elsewhere will protect against the threat of the captured terminal gaining access to the computer and causing system damage. However, it would be possible for the captured terminal to tap the communications link between two other terminals in the same cryptographic net and thereby receive any classified information passed over the link by either of the other terminals. This threat exists in any secure communi-

*Spoofing: intelligent deception

cations system where there are many holders of the same key. It can only be countered by keeping the number of holders of the same key to a minimum and the ideal situation is to have no crypto nets of more than two holders. Crypto-equipment could be located at the computer facility which stored all the individual independent keys of each terminal upon demand. Equipment with this capability does not now exist but is within the current state-of-the-art and can be provided, possibly by modification of equipment now in development but certainly by new development.

2.3 Software Controls

2.3.1 Access Control

Given that the identity of the user has been verified and his console identified, the software system can look up the user clearance and that of the terminal (console) in system directories. The system now has the "user clearance", the "terminal clearance", and the clearance of any other I/O devices accessible to the user's program such as tapes drives, readers, printers, and other consoles.

2.3.1.1 Limiting Input and Output - Software traps should detect any inputs which are identified as having a classification "greater" that either the user clearance or the terminal clearance. Input faults should cause a security violation alarm. Similarly, outputs to any device which are classified "greater" than the terminal clearance or "greater" than the user clearance should be omitted and the fault logged.

2.3.1.2 Limiting Job Classification - The actual job run centrally in the computer should not be allowed to access information (programs and data files) which has a classification "greater" that the user clearance.[1] This limitation is dictated by the judgment that it is almost impossible to guarantee that a job cannot somehow downgrade information so as to permit it to be output. However, exceptions are possible; e.g., an execute only program which has been certified to always digest information so as to produce an output of lower classification than the source data.

---

1. System access to accounting and control files are excluded from this restriction.

2.3.1.3 The job clearance is not limited, however, by the terminal clearance. A TOP SECRET cleared person might run a TOP SECRET job using a SECRET terminal if there were no TOP SECRET I/O to the console. He might direct the output to TOP SECRET printer. Such occurrences might be common for remote initiated batch operations and no deception need be suspected since the user is cleared for the job.

2.3.1.4 Access Control Matrix - The previous access control limitations can be represented by a control matrix. The matrix should be read in the form: User (device) clearance should be ≥ to input (job, output) classification.

Clearance:	Input Classification	Job Clearance	Job Classification
User	≥	≥*	≥
I/O Device	≥	Independent	≥

*Except for certified execute only programs

2.3.1.5 Denial of Access - The "User" who interacts directly with the system must not be allowed access to information related to the character of the control systems and/or files when access is denied under authentication procedures. For example, responses during authentication that provide such information include the following: "You do not have the required clearance to access file X" or "You have not requested file X correctly" or "The classification of file X is Y". A record of authentication failures will be included in the system log and alarms will be provided to assure that the security of the system is maintained.

2.3.1.6 Classification - Clearance Comparison - Section 3 contains a description of a language which informally describes the structure and rules of clearances, special accesses, compartments, classification, etc. Using a language of this form, the definitions needed for a particular installation can then be compiled, creating a data structure. Utilizing this data structure, compare and combine routines can make the somewhat complex determinations of whether a particular clearance grants access to a classification and what "high water mark" classification results from accessing information at several different levels of classification.

2.3.2 Restricted Operations

The threat to the security of the system is reduced in those cases where the allowable functions of specific terminal or I/O devices are limited by the system. These properties can be used, after adequate certification, to support operations and procedures not otherwise available to unrestricted devices. An example of a restricted operation may be a terminal limited by the system to the execution of one certified program that performs some limited set of well defined operations and provides output to another I/O device. In this example the terminal operator could execute the program, without clearance for the data being processed. Another example could be to limit a terminal to a specific set of input functions only, in this case the operator could again operate at a lower clearance level than the files being referrenced.

2.3.3 Output Classification

To assist the user in protecting sensitive information, when a file is created the computer will define the maximum composite classification of the file to the user. This "high water mark" will be determined algorithmically by consideration of the classifications of all files referenced, programs utilized, and inputs utilized.* In case of a permanent file, the user will be notified of this "high water mark" classification. He will then be required to state that he certifies this classification, upgrades it or downgrades it. If a downgrading (a less stringent classification) is certified, the system should audit this transaction for periodic review. Possibly, downgrading authorization should be limited to those users with write access to a file.

The reason for requiring the user to confirm or modify the computer determined classification rather than specify his own is that the user may not realize the totality of all file classifications referenced and most likely has not reviewed the totality of the resultant file. Any analogy with documents handling rules is dangerous since files are often large and non-textual for which the user must determine the content by program testing and/or scanning.

2.3.4 Write Controls

Write access to files must be a function of the specific files and the user. The write capability is not related to security specifically, but is primarily a management function to provide file protection from undesirable changes. The system must provide for these controls in the file structure. The same controls should also apply to deletion of files. The application of such controls increases the integrity of files.

3. Automation of the Multilevel Security Classification and Control System

3.1 INTRODUCTION

This section describes a language that would permit a formal definition of the structure of the U. S. security system for control of classified information. It is a working paper intended to convey initial results in the area under discussion and, as such, is quite informal. It is written for people already quite familiar with the security system. A more detailed version of this work will be needed for the general reader attempting to do the actual implementation.

The basic multilevel security problem consists of answering the following question: Can an individual with a particular clearance have access to a quantum of classified information in a given physical environment? While this problem exists independent of computer systems, the introduction of an automated decision process requires a very formal specification of the decision roles. This section addresses itself to one solution of that problem.

Because of the complexity of the overall scheme for controlling access to classified defense information, it may well be that at no installation is the full range of the security control apparatus necessary. Furthermore, as a matter of precaution, it would be undesirable to divulge unnecessarily to programming personnel the details of security control methods. Therefore, our approach has been to conceive a scheme in which

only the structure of the security control procedures will be described to programming personnel of a nonical clearance level (SECRET). However, there are additional parameters of security control which are not available at the clearance level of SECRET. Our scheme is such as to permit a local security officer cleared to an appropriate level to insert such supplementary information.

We propose to keep in a multi-access remote-terminal computer system, the following information:

1. For each user a list of certain parameters relevant to him.
2. For each file a listof certain access parameters relative to the information contained in that file.
3. For each terminal connected to the system, a list of certain parameters relevant to it.

The details of these parameters and how they are used is developed below.

We have made certain assumptions and definitions as follows for the purposes of this paper.

1. For whatever part of the total security control system that may be of concern to a given installation, the security officer (or other responsible party) at that installation must know the complete structure.
2. The need-to-know control on access will be verified by explicit reference to a name check, organization check, other check, or combination of checks, etc., as may be required by security procedures. This is in addition to

verification of the clearance status of the user requesting access to a given file.

3. A clearance is associated with either a user or a terminal; a classification is associated with (a file of) information.

4. The word "accesses", when used below as part of the security structure language, is defined to be semantically equivalent to "permits access to information labelled as".

5. The phrade National Clearance is taken to mean the normal defense clearances of TOP SECRET, SECRET, CONFIDENTIAL and UNCLEARED, which are hierarchical in that order. The National Clearance status of an individual will be taken as the major parameter in controlling his access.

6. If an individual is authorized to have access to information of Type A at one or more National Clearance levels, then it is assumed that he is (in principle) granted access to Type A information up through the level of his National Clearance. This is intended to rule out the following case which we believe is common in present manual practice: an individual with a National Clearance of TOP SECRET is authorized access to (say) cryptographic information (i.e., is granted CRYPTO access) only to the SECRET level.

We regard this as an illegal use of the clearance control structure and for the purposes of the computer records, an individual granted (say) a National TOP SECRET Clearance and access to information of Type A, is automatically assumed to be cleared for all Type A information through the TOP SECRET level. Any exceptions to this assumption must be explicitly stated on an individual basis. Thus, it can be said that the National Clearance factors or distributes over all special information types. The phrase "Type A" can refer to a special clearance system, a compartment or special grouping which itself may be within a special clearance system, or any major or minor segment of any clearance system that may have to be specified.

7. As a consequence of the above, the computer algorithm which matches parameters of the user against the parameters of the file to be accessed will compare the user's National Clearance and the file's National Classification first. If a user is to be granted access to a given file, then his National Clearance level must equal or exceed the National Classification level of the file. Note that this is a necessary but not sufficient condition for access. Additional control such as code words, special access, compartments, special groupings, etc. will be regarded as controlling access to specific information types within the framework of the National Clearance structure.

8. An "Access Control Label" is regarded as an additional means of access control and will require verification against the user's status. Examples of such labels are: "No Foreign Dissemination except Intelligence Elements", "Not Releasable Outside the DoD".

9. An "Informational Label" is regarded as not controlling access to the information but rather giving guidance to the user on how the information may be further disseminated, further controlled, utilized, etc. Examples of such labels are: Limited Distribution, Special Handling Required.

10. All names, code words, etc., at a given installation are assumed to be unique.

SYSTEM CATALOGS

The computer system will maintain a catalog of all terminals which may be connected to it and for each terminal will maintain the following information:

1. The highest level of National Calssification of information that may be transmitted to or from the console.

2. Special code words, group names or other names which modify the National Clearance level of the console to receive other classes of information.

3. Physical location, including building location, room number and the cognizant agency.

4. The electrical address.

5. The permenent identification number.

6. A list of the user authorized to use the console.

7. Person responsible and his telephone number.

The computer system will maintain a catalog of all users authorized to have access to it and for each user will maintain the following information.

1. His National Clearance level and its date of expiration and granting agency.

2. Special code words, special groupings or other words which extend his access to other classes of information, and the date of expiration of each such special name.

3. His agency affiliation.

4. His citizenship.

5. His agency assignment(s).

6. His permanent identification number (Social Security or other).

7. Special need-to-know designators other than those explicitly given by items 1 and 3 above.

The computer system will maintain the following information for each of its files.

1. The National Classification of the file.
2. Special names such as code words, compartment names, handling caveats, etc. that serve to control access to the file.
3. A need-to-know list, including one or more of the following as may be required.
   - Universal need-to-know; i.e., everyone authorized access.
   - A name list.
   - A group designator need-to-know.
   - Specific exclusions from need-to-know by such things as groups, names, explicit lists of names.
4. Access control labels.
5. Informational labels.
6. Background information on the file. (This is subject to policy decisions.) Examples of information which might be desired are:
   - Its date of creation.
   - Its downgrading group, and any downgrading actions applied to it.
   - Name of individual who created the file and his agency.
   - Predecessor files (if any) from which the file was created.

3.2 SECURITY STRUCTURE

Not only will the computer system maintain a catalog of information for each user, each file, and each terminal but it also must be aware of the structure of the security control apparatus. This paper defines a special language in terms of which the security structure can be stated, and for which software will have to be written for each machine and/or software system that is placed into operation.

The security structure language described below formally defines a set of relations between entities. These entities, include special accesses, code words, etc. The structure below can be though of as defining a set of decision rules which the computer system can then consult when it wishes to make a decision concerning security parameters. It is immaterial as to how these decision rules are actually stored in a computer, and this is (for the present) left to the individual software system designers.

To define either a major or minor element of the security structure, we propose the following syntax. The traditional notation and format of a programming language is used. The default condition of definition is mutual independence in that, unless otherwise indicated, two defined entities will be assumed mutually independent.

ATTRIBUTE	NOTES
Define Name	(1)
Clearances:	(2)
Synonyms:	(3)
Required Labels:	(4)
Structure:	(5)

ATTRIBUTE	NOTES
Access Rules:	(6)
Relational:	(7)
END	

NOTES:

(1) "Name" is the word which acts as a label to identify the security element to be defined.

(2) The names of the clearance(s) which exist within this security element.

(3) Any commonly occurring synonyms or abbreviations of names of clearances, labels, etc.

(4) Any required labels which must be associated with the information to which access is controlled by the security element being defined. There is a special reason for these labels which will become clear in an example below, but it is assumed that all of the clearances of the security element being defined are permitted access to information labelled by one or more of the required labels. Of course the other access criteria must also be satisfied.

(5) Structure information is regarded as being totally internal to the security element being defined. Within

the "Structure" part of the syntax there is only one functional operator called "Imply". This is interpreted to mean that access authorized by a given clearance implies the automatica access (unless otherwise limited) authorized by other clearances lower in the hierarchy. If an individual has a TOP SECRET clearance, "TOP SECRET" implies "SECRET" in the sense that an individual cleared TOP SECRET has access to information to which an individual cleared SECRET also has access.

(6) Under "Access Rules" there is only one operator called "Accesses" which has been previously defined as "permits access to information labelled as". These rules explicitly state the relation between the names of the clearances in the security element being defined and the labels on the information to which this security clearance permits access. In many cases, the same word is used to specify a clearance and as a label to indicate classification of information.

(7) The "Relational" information establishes any links between the security element being defined and other parts of the security structure; thus, it is regarded as external to the element being defined. Two operators are allowed: (1) the "Imply" and (2) "Requires". The "Imply" means the access granted by the security element being defined

automatically denies or authorizes access (unless otherwise limited) to stated other categories of information. "Requires" provides for the case in which access to information labelled by the security element being defined requires the simultaneous existence of a particular other clearance or access authorization. Boolean expressions are allowed for both operators.

3.3 Examples

Several examples are given to illustrate the use of the security structure language syntax. The subsection may be safely skipped by the casual reader.

Let us define an element of the security system whose name is "National Clearances", which contains the clearances TOP SECRET, SECRET, CONFIDENTIAL, and UNCLEARED, for which no special access labels are needed, which has the hierarchical structure that a TOP SECRET clearance implies a SECRET clearance implies a CONFIDENTIAL clearance implies UNCLEARED, and in which access is (in three of the four cases) controlled according to the rule that a particular clearance may access information labelled with the same word; e.g., a SECRET clearance authorizes access to information labelled as SECRET. In our system, then, it would be done as follows:

Define: National Clearances

Clearances:
TOP SECRET
SECRET
CONFIDENTIAL
UNCLEARED

Synonyms:
TOP SECRET-TS
SECRET-S
CONFIDENTIAL-C
UNCLASSIFIED-US
UNCLEARED-UR

Required Labels: None

Structure:
TS implies S
S implies C
C implies UR

Access Rules:
TS accesses TS
S accesses S
C accesses C
UR accesses US

Relational: None

END

Let us now consider the base of cryptographic information as discussed earlier and define a class of information called "CRYPTO" which is to be regarded as a further restriction on access under the National Classification System.

Define:	CRYPTO
Clearances:	CRYPTO
Synonyms:	None
Required Labels:	Handle via special channels
Structure:	None
Access Rules:	CRYPTO accesses CRYPTO
Relational:	CRYPTO requires TS .OR. S
END	

This example illustrates the use of a label as an access control. It has been assumed that CRYPTO information is to be transmitted via special channels. However, there may be administrative traffic which will not have the classification label CRYPTO, but access to which must be confined to CRYPTO-cleared people. Thus, TOP SECRET or SECRET information carrying the special handling label assumed in this example can be identified as CRYPTO-class information and access controlled accordingly. In effect, a required label can be regarded, when necessary, as a pseudo-clearance, accessed by any of the clearances listed in the definition. The notation .OR. identifies the Boolean operation of disjunction.

Of course, it would have been sufficient just to put S since TS implies S.

Let us define next a special class of information called Restricted Data which is assumed to exist within the National Classification structure.

Define:	Restricted Data
Clearances:	RESTRICTED DATA
Synonyms:	Restricted Data-RD
Required Labels:	None
Structure:	None
Access Rules:	RD accesses RD
Relational:	RD requires TS .OR. S

END

Here we note that there is only one way to identify information that belongs to this element, and that is through the use of the label RESTRICTED DATA. Yet, the access rule is necessary to specify the fact that TD is both a clearance and a classification (label). Note again the use of .OR. in the relational statement; the Boolean operators .OR., .AND. and ,NOT. should all be allowed wherever needed in the syntax.

Let us now define a hypothetical clearance called DATATEL having three clearance levels within it referred to as III, II, and I. DATATEL information of clearance level III carries the code word ABLE; II carries the code word BAKER; and I carries the code word CHARLIE.

Define:	DATATEL
Clearances:	III, II, I
Synonyms:	None
Required Labels:	Handle via DATATEL channels only
Structure:	III implies II implies I*
Access Rules:	III accesses ABLE II accesses BAKER I accesses CHARLIE
Relational:	III requires TS II requires S I requires C

END

Let us now define a compartment of information within the DATATEL structure whose name is APPLE, and for which the label ALICE is used to identify information contained in this special grouping.

---

*This is an alternate notation to that used in the first example. As in the first example it could also be written:

III implies II
II implies I

Define:	APPLE
Clearances:	APPLE
Synonyms:	None
Required Labels:	Handle via APPLE channels only
Structure:	None
Access Rules:	APPLE accesses ALICE
Relational:	APPLE requires III
END	

This illustrates a variation possible in this syntax. It has been assumed that APPLE information is not labelled as such, but is to carry the additional label ALICE. The APPLE definition relates APPLE to III; the earlier DATATEL definition relates III to ABLE and also to TOP SECRET. Thus the system can correctly determine that the proper label for APPLE information is TOP SECRET ABLE ALICE. A required label of ALICE need not be given since the access rule contains the information.* We now observe something else about required labels. APPLE information would have two required labels (in this case, two

*The examples have contained only handling rules as required labels. In general, it is suggested that information not be given twice in a definition. For example, the compiler can acquire the ALICE label from the access rule. If other kinds of information must be specified as required labels, then some rules about the format of information in the required label block will have to be developed.

handling requirements), and logical rules have yet to be established to handle such situations.

Let use define an example in which it is assumed that at the SECRET level there are two categories of information called AGILE and BANANA accessing information labelled respectively as ANN and BETTY. Further assume that an individual cannot be concurrently authorized access to AGILE and BANANA information. To have access to both, assume that an individual must be cleared TOP SECRET, in which case he will be said to have access to CHERRY information labelled CHICO, as well as all AGILE and BANANA information.

The specification of these three security elements will then be as follows:

Define:	AGILE
Clearances:	AGILE
Synonyms:	None
Required Labels:	Handle via AGILE channels only
Structure:	None
Access Rules:	AGILE accesses ANN
Relational:	AGILE requires S .AND. .NOT. BANANA CHERRY implies AGILE .AND. BANANA

END

Define:	BANANA
Clearances:	BANANA
Synonyms:	None
Required Labels:	Handle via BANANA channels only
Structure:	None
Access Rules:	BANANA accesses BETTY
Relational:	BANANA requires S .AND. .NOT. AGILE CHERRY implies AGILE .AND. BANANA

END

Define:	CHERRY
Clearances:	CHERRY
Synonsyms:	None
Required Labels:	Handle via CHERRY channels only
Structure:	None
Access Rules:	CHERRY accesses CHICO
Relational:	CHERRY requires TS CHERRY implies AGILE .AND. BANANA

END

Note that in these examples Boolean operators have been used in the "Imply" statements. With reference to the definition of AGILE, for example, the relational rules state that AGILE access requires SECRET National Clearance and not the concurrent access to BANANA information, and that CHERRY access implies automatic access to both AGILE and BANANA. While access control to the AGILE compartment would not normally require the statement that simultaneous access to AGILE and BANANA requires CHERRY access, nonetheless this relation is given as part of the AGILE and BANANA definitions to facilitate the automatic assignment by the computer system of access controls to new files which a user may create from old files. Thus, should a user merge AGILE and BANANA information to create a new file, the merging algorithm in checking for access controls to be applied to this new file would, in consulting the definition of AGILE and the definition of BANANA, discover that the simultaneous presence of both require a CHERRY access control on the new file. Then in consulting the definition of the CHERRY element, the merge alforithm would discover that CHERRY requires TOP SECRET, and hence would label the new file as TOP SECRET CHICO. (Note in this particular example that the label on a file and the clearance required to access it are different. In other cases the same word might be both a label and a clearance.)

3.4 General Discussion

In general, it is believed that the file merge algorithm probably should assign to a new file the highest National Classification of the set of National Classifications attached to the information from which the new file was assembled, and further, that it should concatenate any special names to be applied to the new file, subject to any exclusion rules which may exist among them.

The algorithm which compares the parameters of a user requesting access to some file with the parameters carried in the header of that file is visualized as first checking the National Clearance and National Classification involved; then any special compartment names, special group names, or other names; then any restriction which the required label may impose on access; then any special access designators which may exist, but which are not explicitly identified as a label; and finally verifying access by a name check.

It is believed that the mechanism which has been outlined above will suffice for the description of all parts of the security control system. In a document at this classification level, it is not possible to include the examples which demonstrate that it does work in all places, but it has been checked that it is adequate for all those systems about which sufficient knowledge is available. On the other hand, exhaustive knowledge of all details of the entire security control structure is not claimed, and it is possible that pathological cases exist which cannot be described in the language.

It should be noted that the actual dynamics of the system have yet to be formally specified. That is, the programming algorithms for all relevant decisions are not yet formally specified. That is, the programming algorithms for all relevant decisions are not yet formally defined, since this is the basis for future work. For example, it appears that the access algorithm would examine the "Requires" statement in the Relational section prior to the "Imply" statement, while the merge algorithm would proceed in the reverse order. The latter must be the case in order to avoid tricking the system into thinking it had discovered a logical inconsistency because a CHERRY cleared person had accessed both AGILE and BANANA information, the clearances for which cannot be mutually coexistent.

Security is not the only issue which concerns access control to files. While the present section deals only with the security aspect, related problems are mentioned here because the software procedures will have to be designed to deal with all aspects of the problem.

One related problem is that of file management. Even though a given user may have the clearance which authorizes him access to a particular file, in the interests of controlling who takes various actions on the file, the file management system may not grant him access or may limit what he can do with the file. This can be looked on as a form of need-to-know and can, in principle, be dealt with by keeping several need-to-know lists with each file. For example, authority

might be granted for:

1. Reading only of the file.
2. Changing existing specified fields.
3. Adding new entities or new fields.
4. Purging the file by deleting old entities or fields.
5. Creating a new file from the given file.

This need not beseveral lists but could in reality be one list in which each name carried authority codes with it. On this point, the file management problem and the security problem intersect.

File creation is a second problem not discussed in this work. Many conditions might have to be accounted for, some of which are:

1. A file might be duplicated which implies that the complete header information of the original file would also be copied.
2. Concatenating two files with elimination of no entities which implies that the two headers will be joined according to some merge/replacement algorithm.
3. A file may be partially copied, which implies that (probably) only part of the header information from the original file is relevant to the new file. It might be necessary to add new header information to the second file on the basis of such things as:

a. Whether the new file was created by a program that has been cataloged and certified, or

b. Was created by a program that is experimental or in debug, or

c. Was created by a program that was finished but uncertified.

4. A file may be duplicated by simply renaming

Partial copying could result in lowering of the classified level of the file and, thus, the header information will have to be modified. It may prove possible to design logic which will handle such downgrading automatically, but certainly many cases will have to be determined by the security officer.

This section does not discuss in detail the various ways in which the catalog of user, catalog of terminals, and file header must be manipulated. For the record, the following (not necessarily complete) list is given.

1. A user clearance and the terminal clearance at which he is presently working must be properly conjoined to establish the present job clearance.

2. Job clearances and file calssifications must be compared to control access.

3. File header information, particularly of internally created files, must be utilized to properly label printed or displayed information. This includes not only national classification, but also special compartment or group names, informational labels, and other required labels.

4. File header information in conjunction with other logic must be used in the automatic assignment of classifications and the automatic generation of headers for new files.

At present, no provision for classifying the deck of cards (or whatever) containing the description of the security structure for a given installation has been included. This is because this information can have a classification outside of the structure it defines. Rather, this information is considered to be so sensitive that its access must be controlled on a specific name authorization only. Thus, when this information is resident in the computer system, its access must be controlled by a special purpose mechanism not part of the regular file system. In general, this attitude is adopted toward all of the critically sensitive portions of the software system.

1. The central computer system, all remote terminals and communication lines are physically secured according to already-established regulations and precedents for providing such physical security;
2. All personnel who have access to terminals or the central facility have a security clearance acceptable to the responsible officer;

These procedures should apply at any time that there are information-storing devices which may contain classified material physically connected and accessible to the operating system. Provision may be made for switching off, removing or purging such storage devices. It is then possible for the computer system to be operated without the above assumptions in force, providing that all classified information is physically inaccessible during such operation.

For a system which proposes to provide privacy, it is presumed that the certification that it actually does will be done by the official who is responsible for the security of all of the information which is available through the system. This section is intended as a guide which may be used by that official in making his judgement, and by a supplier of a computer system hoping to achieve certification. It is recognized that no currently available computer system in fact meets all the suggested guidelines. To the extent various features are not avail-

4. STATE-OF-THE-ART PRIVACY SYSTEMS

4.1 Introduction

The purpose of this section is to provide guidance to groups which need to install a time-shared computer system to process classified information, but based on state-of-the-art understanding of measures to provide information protection. It is apparent that the general problem of providing security for classified information stored in a time-shared computer system is currently unsolved. However, in view of the evident interest in providing time-sharing capability for the processing of classified information, there is a need to suggest such considerations and guidelines as may be practicable at the present state-of-the-art. These guidelines are provided so that one may determine to what extent a proposed system may be adequate for a particular application which does not require full multilevel security procedures.

The following guidelines take the point of view that currently available system construction techniques are potentially capable of providing what is technically termed privacy. A system providing privacy, by definition, provides safeguards against releasing classified information to individuals who are cleared for the information, but have no need-to-know. In such a system, security is presumed provided by mechanisms external to the computer system itself. For example, the following two procedures might be used:

able management controls and manual features may be substituted if the resultent restricted operation can be accepted. Such lack of achievement should not necessarily prevent certification, but it should indicate to the certifying officer the risk he is taking if he accepts the system.

The comments in this section apply to a generic class of computer systems, commonly referred to as "time-shared", which have all of the following properties:

1. They are based on a general-purpose digital computer.
2. They store information (programs and data) on a long-term basis for the users of the system. The system takes on the responsibility for reliability of storage as well as insuring that stored information is not compromised.
3. The system provides simultaneous access for several users using techniques commonly referred to as "multiprogramming", "time-sharing", or "multiplexing", which distribute common resources (e.g., central processor and primary and secondary memory) among the several users according to instantaneous demand.

In addition, the system may be accessible from a distance by a typewriter or other terminals connected to the system by communication lines. Such systems may supply differing services to users which present progressively more difficult environments in which to realize a privacy system, e.g.:

1. A system which permits the user to execute only programs

provided as part of the system.

2. A system which interprets a user-provided program.

3. A system which permits direct execution of only programs generated by a system-provided compiler.

4. A system which permits direct execution of any user-provided program.

We concentrate our attention on the fourth (and most difficult) form of service (although the line between 3. and 4. above is often difficult to draw) because the current need for time-shared privacy systems appears to extend to such service.

To afford privacy, such a system must provide an authentication mechanism by which users of the system may be appropriately identified and a protection mechanism by which identified users are given access only to information to which they are entitled. Furthermore, the system must be constructed in such a way that imely certification by competent authority is feasible.

4.2 Authentication

Authentication is the means by which the computer system is assured that the individual at a terminal is who he represents himself to be. User authentication is usually provided on existing systems through a pass-word. This technique can provide adequate protection for privacy purposes if:

1. The pass-words are given protection comparable to that required for the most sensitive information available to that user;
2. They are changed periodically to minimize potential loss (comparable to changing safe combinations);
3. They are not user-generated (to prevent penetration by educated guessing).

More elaborate schemes such as one-time pass-words or challenge-dependent pass-words may not be necessary to achieve the objectives of privacy. However, installations handling sensitive material or attempting to approximate secure environments should require them.

4.3 Protection

To provide protection of information stored in the system, certain hardware features can be described as essential for a system which allows execution of user-specified machine language instructions. These same hardware features can also help simplify certification of systems which do not allow machine-language programs, although supervisor procedures can in some cases can be provided as a substitute.

1. The execution state of a processor should include one or more variables (the protection state variables) which determine the interpretation of instructions executed by the processor. (For example, a processor might have a master mode/slave mode protection state variable, in which certain instructions are illegal except in master mode.)
2. Modification of the protection state variables can only be performed under circumstances in which control of the process is simultaneously transferred to a procedure qualified to operate in the new protection state. (For example, an interrupt may switch the protection state to master mode and simultaneously transfer control to a supervisor provided interrupt handler. When the handler completes its operation it may explicitly restore the old perfection state, as well as the program formerly in control.)
3. The ability of a processor to access locations in primary memory should be controlled on a permission basis which may depend on the protection state of the processor. (e.g., in slave mode, a memory permission register might allow access only to primary memory locations belonging to the user in control.)

4. The correct operation of certain instruction should depend on the protection state of the processor. (For example, instructions which indicate input or output operations on a shared storage device (e.g., disk or drum) would execute properly only when in master mode. Any attempt to use such "privileged" instructions in slave mode should cause interruption of the program containing the instruction.)

   Note that it may be acceptable if the user can execute input/output instructions directed to devices assigned exclusively to him.

5. All possible operation codes, with all possible tags or modifiers, whether legal or not, must produce known responses by the computer.

The system software should utilize these hardware features to limit access to data to authorized users. In particular:

1. Any violation of memory bounds or attempted execution of privileged instructions should cause monitor action to log entries, and a reasonable time delay before the user may continue execution. This time delay should be long enough to discourage methodical probing. Provision should be made for the security officer to deny access in suspicious situations.

2. The monitor should be organized in such a way that it is not necessary to suspend security procedures in order for users to debug their programs.
3. Procedures should be available for clearing from the system (or making inaccessible) all classified information during actions which must be run without the normal protection.
4. The monitor should insure that sensitive data does not remain as accessible residue in primary memory or on secondary storage devices.
5. The monitor system should include procedures for an orderly shutdown of the system when desired.
6. Logs should be kept of the highest level of classification of information which has ever been stored on a device, so that disposal and decontaimination policy requirements can be met.
7. System data bases and tables should be interlocked when updated in such a way that access to a table is prohibited whenever other tables are not consistent with it. For example, adding or deleting a user of the system may require modifying several tables. Interlocking techniques should be used which insure that an attempt by the user, say, to log in during his own deletion, will be delayed until the deletion is completed.

8. Supervisor data bases should have consistency checks in them which are routinely checked whenever the data base is referenced. For example, a string pointer list might include both forward and back pointers.

9. Procedures should be provided to adequately protect duplicate copies of stored files as well as the originals. For example, if files are copied onto magnetic tapes for backup, the tapes must be guarded as carefully as the on-line information storing devices.

In addition, to ensure that hardware and supervisor information protection features are operating correctly, the design of the system should include provision for automatic periodic testing of protection features. For example, periodic tests might include:

1. Verifying sensitive portions of the monitor (e.g., the security tables) against master copies for possible change.
2. Generating unauthorized addresses or privileged instructions in user mode to insure that protection hardware is working.
3. Verification that less frequently used features which the supervisor depends on (e.g., privileged instructions or time of day clock) are operating properly.

4. Verification that supervisor data base consistency checking features are working correctly. This verification can be done by temporarily damaging the consistency check data, and exercising the appropriate supervisor procedure to see if it notices the damage.

5. Periodic comparison of counters kept by the supervisor with counters maintained by the hardware of the number of read or write requests issued to information storing devices.

6. Error detecting and correcting techniques may be useful in assuring correct operation of devices used to speed up the processor such as associative memories, instruction stacks, or look ahead registers. In any case provision should be made for the supervisor program to verify their correct operation. For example, one should have the ability to turn speed-up hardware on and off by privileged instruction, and to store contents of related registers to verify correct operation.

Since it is common on a time shared system for the programs of the supervisor to be maintained on the system itself, certain special precautions must be taken regarding these programs, in both their source and executable form:

1. It should be possible to separate the authority to modify the supervisor actually in use from the authority to debug a proposed modification. In other words, the authority to obtain a copy of a supervisor procedure, modify it, and test it under special operating conditions may be vested in a number of system programmers; the authority to install such a checked out modification as part of the working system routinely offered to user may be vested in a single person.
2. The source program of the system should be protected against changes by unauthorized programmers, since a later authorized change, if installed, would also introduce the unauthorized one. One procedure to insure that unauthorized changes do not occur is to allow access to modify the system master copy of a source program only to a person authorized to modify the actual running system. A system programmer would debug proposed changes on a copy of the master, not the original. When he is satisfied with his changes, he

turns a list of changes over to the person responsible for introduction of changes into the working system. These changes are edited into a new copy of the master source program, which is then compiled, tested, and then introduced on the new master.

4.4 Certification

In the process of certification, the combination hardware/software systems is to be subjected to inspection and test by expert technical personnel to determine the degree to which it conforms to the requirements of appropriate regulations and policies. The extent and duration of the inspection and testing is left to the discretion of competent authority and will depend heavily on the manner in which the hardware and software is constructed. In order to keep the certification period to an acceptably short period of time it is advisable to follow certain practices in constructing the system, i.e.:

1. Reasonable (an ill-defined term) hardware reliability features, such as memory parity checking hardware, etc., should be provided so that the other safeguards of the system can be assured to be operating correctly. In this regard, unduly complex hardware design will complicate the certification process, as it will cast doubts both on its reliability and the integrity of the hardware protection mechanisms; therefore:

1.1 The use of complicated schemes which make the operation of instructions potentially erroneously dependent on the operation of adjacent instructions should be avoided. Features such as look ahead or pipe line organization require very careful analysis to determine if they are free from this class of fault.

1.2 It is advantageous to use a standard addressing mechanism so that the memory protection mechanisms are independent of machine instruction operation code decoding.

1.3 Asynchronous organization of hardware logic provides hardware interlocks against unexpected delays caused for example, by component drift or misadjustment. Such organization therefore, can ease certification.

2. Other hardware features which would enhance the certifiability of the system include:

2.1 Program readable hardware configuration status switches, thus insuring that the software is aware of the hardware configuration in which it resides. If it possible to set up illegal or inconsistent configurations, there must be available a program which can detect such illegal or inconsistent settings. In particular, a test to insure that all maintenance switches are in their normal operating positions should be provided.

2.2 Provisions to control unauthorized or accidental changes to configuration and peripheral control switches. A key lock on a control panel cover is an example.

2.3 Program readable clocks which provide date and time of day for use in controlling audits and recording file creation instants.

3. As the monitor will, in the large part, be produced by uncleared personnel, it will be necessary for the certifiers to assure themselves that no "trapdoors" (intentional or unintentional) for unauthorized access will have been introduced into the system. This certification may require examination of the operating system code on a line-by-line basis by certification authority. Therefore:

3.1 Use of esoteric coding techniques is to be discouraged.

3.2 Use of higher order programming languages where possible is important. Since the compilers for such languages are potentially capable of introducing such trapdoors into their object code, the compilers must themselves be certified and should be written in (their own) higher order language if possible.

3.3 The monitor should be constructed in a modular fashion such that errors occurring in one module do not affect the internal operations of another. As much of the monitor as possible should operate under the memory protection scheme

(as opposed to supervisor mode). It is advantageous to segregate portions of the monitor which need not deal with security matters.

3.4 Data should be separated from the instructions of the program, as to simplify protection the instructions with hardware facilities and insuring that fresh internal storage areas are used when the program is reused.

4. Good documentation of both the hardware and software is essential.

4.5 Summary

1. A general solution to the problem of either security or privacy in time-shared computer systems is not currently within the state-of-the-art.

2. Selective certification of such systems for specific privacy applications in the near future seems possible. The features that such systems should possess can be identified; systems possessing such features are currently in operation.

3. A critical problem in the near-term utilization of such systems is the potentially excessive time required to achieve their certification for particular privacy applications. This section attempts to indicate features these systems might offer to reduce this time to acceptable proportions.

5. SECURE SYSTEMS

5.1 Introduction

By Secure Systems, we mean systems that permit uncleared users using unsecured lines to share a time-shared facility with cleared users running classified problems. We further assume that the time-shared facility exists in a physically secure environment, and incorporates all of the applicable methods discussed in previous sections.

The crucial difference between the security problem and the privacy problem is the existence of (1) uncleared users, and (2) unsecured lines. Even with the facility existing in a secured environment, the exposure of the system to uncleared personnel (whether or not operating with "invisible" secured lines) poses a set of requirements on the system that cannot be solved by administrative of policy actions.

The "open" aspect of the system(s) also makes it more vunerable to various threats discussed below.

5.2 Potential Threats and Countermeasures

The following table indicates advertent and inadvertent threats to a system which can be broadly placed in three categories:

1. Recovery of Information
2. Denial of Use of System
3. Intelligent Deception;

and can occur through:

1. Programming
2. Hardware/Software error
3. Physical Access
4. TEMPEST.

The table also indicates some countermeausres to some of these threats, although it is not complete.

Points of Vulnerability/Attack:

A. User - Programmatic Attack

Threats	Countermeasures
1. Recovery of info by:	
a. Misrepresentation	Authentication - User
b. Illegal read/write (core)	Memory bounds
c. Illegal access (files)	Access controls (see section 3)
d. Tampering with operating systems	Access controls (see section 4)
2. Overload (to deny use of system)	User limitations
3. Local User input	Manual checks
	Audit trails
	Alarms on violation

B. Terminal - Physical Access/Machine Errors

Threats	Countermeasures
Alteration of input	Terminal authentication
	Access controls (see section 4)

C. Line

Threats	Countermeasures
1. Intercept	Securing line
2. Tapping to manipulate systems	User/terminal authentication (other checks against User Prog. attacks)
3. System Denial	Line isolation
4. TEMPEST	Line isolation
5. Third Party	Line isolation

D. Secured Facility

Threats	Countermeasures
1. System Personnel (Op/Prog/Maint)	
a. Program/File change	Access Control (see section 4)
b. Program error	Certification
c. Unauthroized call out	Audit trail
2. Hardware	
a. Mag retention	Cipher only
b. Tampering	
c. Logic faults	Diagnostics

As examination of the table reveals, the points of vulnerability of an "open" system are many, and assume the proper working of nearly all of the operating system and the hardware mechanisms provided to assist an operating system to isolate users from itself and each other. Further, the problem(s) of certifying a time-shared system (taken in the sense of both the software operating system and the hardware of the computer) are very complex indeed. Even if a system were certified, a continuing problem would be to re-affirm the certification.

The present state of operating system design is not so advanced that the operating system will remain static for an extended period of time, consequently, it is probable that continuous certification would be required (such that at least each change to the "certified" system would have to be certified).

5.3 Techniques for Protecting a System

In the face of the threats posed by having a partly open system only three courses of action are available: (1) Close the system (2) Render the material unclassified (3) Certify the system to permit running with both cleared and uncleared users, running classified and unclassified programs.

Clearly the simplest course of action is (1) above. However, there exists within the Government and among DOD Contractors, a sufficient number of cases where a single system should serve both cleared and uncleared users. Closing a system is not necessarily the least expensive method of attacking the problem.

Aside from actual declassification, which is not pertainent to this discussion, the only method for transforming classified material so it may be treated as unclassified is through encryption. As a consequence, use of encryption techniques is one method of making a time-shared system secure, even one having uncleared user using unsecured lines. It should be noted, however, that the principle threat countered by use of encryption techniques is recovery of information. Other controls, already discussed, are required to prevent intelligent deception or denial of service. Use of encryption techniques may have an additional benefit - that of reducing the scope of the system certification problem to more manageable proportions.

5.3.1 Encryption Techniques

Three types of encryption, for use in securing the operation of a time-shared system have been identified. They are:

1. Line Encryption.

2. Primary (Computer Operation) Encryption.

3. File Encryption.

The first, line encryption, is presently employed to secure transmission paths carrying classified information. Internal and file encryption techniques are not now in current use.

5.3.1.1 Line Encryption

Objective - To prevent intercept of information being passed over the line and to make it more difficult to introduce data into the system that could result in either false information or manipulation of the system (intelligent deception).

Method - Employ approved cryptographic devices which will provide point-to-point encryption of the information being transmitted.

Requirements - High speed, long-term security, minimum size, power requirements, etc.

Conclusion - Although there exists today communications encryption devices which provide the required security protection; equipments which will possess the desired physical characteristics will not be generally available until the early 1970's. No additional special research requirements have been identified.

5.3.1.2 Primary Encryption

Objective - To prevent compromise through unauthorized access to data residing in primary storage. The objective of primary encryption is to operate a time-shared system with all data and programs residing in primary storage in encrypted form. Decryption for the purpose of executing programs is done as the data/instructions pass from the primary storage to the CPU logic. As data is returned to primary storage, it is again encrypted. This is illustrated in Figure 5.1. Incorporation of this technique will prevent compromise through unauthorized access to data residing in primary storage.

Methods - The encryption system envisioned is independent in operation. A cryptovariable is generated as a function of the user ID; other cryptovariables that may be required are derived from classification of program or data, user address space, etc. Other cryptovariable sets required for operation of a program (i.e., for common programs, and other system programs) are derived or stored with the cryptovariables generated from the users identification. These cryptovariables last only as long as the user is active. If the user returns after any time off the system, a new set of cryptovariables is generated for him. Cryptovariables associated with common and system programs are regenerated when the cryptoperiod of the primary encryption device changes.

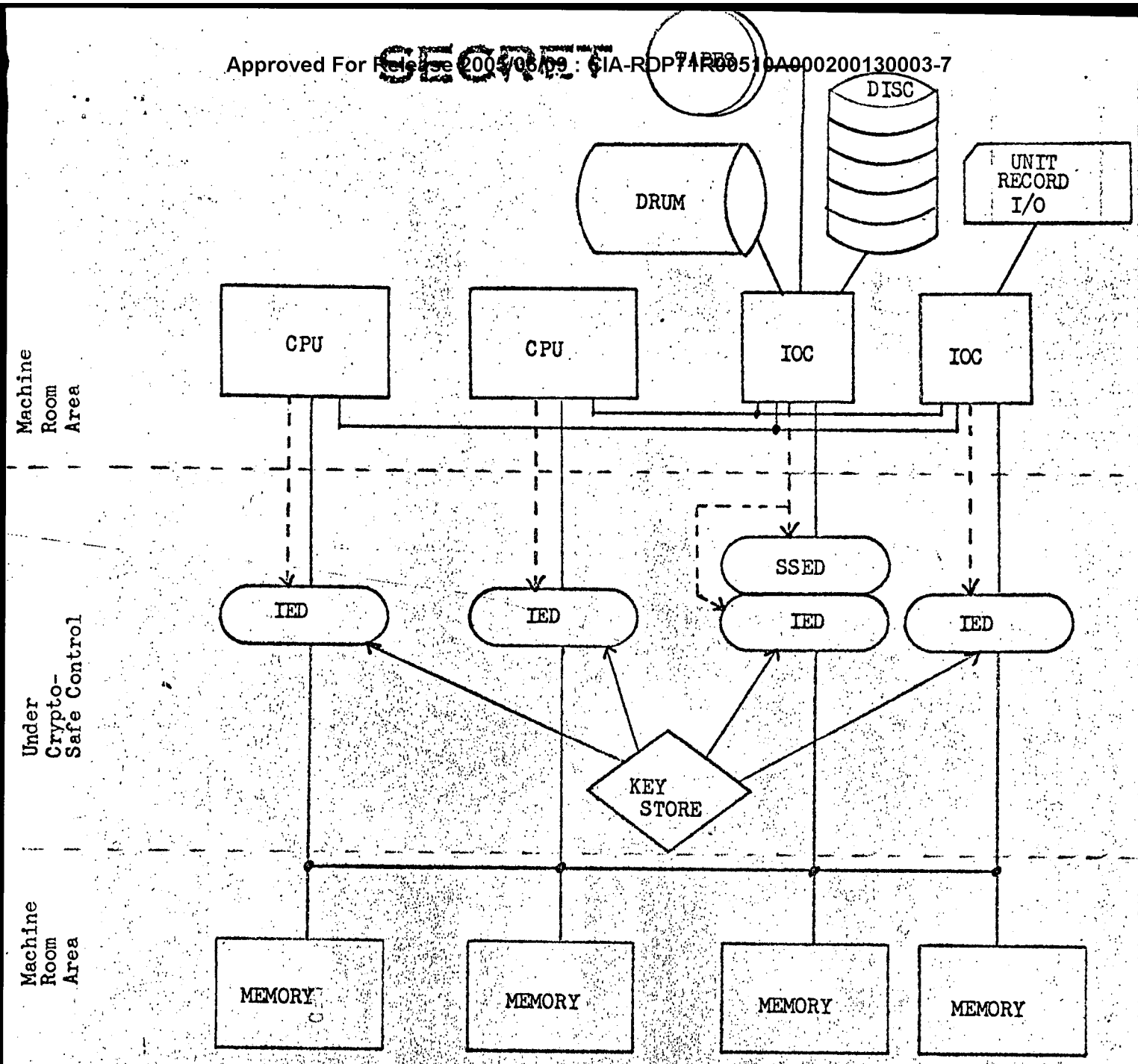

Fig 5.1

SYSTEM DIAGRAM
SHOWING LOGICAL PLACEMENT OF PRIMARY
STORAGE AND SECONDARY STORAGE ENCRYPTION DEVICES

LEGEND:

_____ DATA FLOW
------- CONTROL

IED - INTERNAL (PRIMARY) ENCRYPTION DEVICE

SSED- SECONDARY STORAGE ENCRYPTION DEVICE

The detailed requirements of an Internal (Primary) Encryption Device (call hereafter the IED) for use as outlined above should be developed from detailed system design studies of the nature of such devoces. Without having performed such a study, some of the general requirements of such a device can still be enumerated.

1. Cryptovariables for the IED must be derived from the program execution parameters. The encryption offered each user of a time-shared system must be unique to that user.
2. The IED must operate at memory transmission rates. Interposing the IED between a CPU and the primary storage must not slow the system due to the operation of the IED.
3. The IED, line encryption devices and the Secondary Storage Encryption Device (SSED) must operate together in transmitting information to and from primary storage such that the information never appears en claire. This requires that all of the encipherment devices used to secure a time-shared system operate as transencipherment* devices.
4. IED encipherment requires a cryptoperiod that is at least on the order of several hours, and must be suitable for use with a very large number of short, probably fixed length, units of information. The short term characteristic is a reflection of the nature of a time-shared system.

---

*transencipherment: The application of a pair of (possibly different) cryptographic transforms in such a way as to convert ciphertext enciphered under one transform to ciphertext enciphered under the other transformation such that no immediate plaintext results.

5. Separate cryptovariables are required for each address space accessible to a given user. In modern machines, address spaces are accessed through base register relative addressing. The general rule applies that each active base register has at least its own cryptovariables associated with it. If the IED technique is applied to provide an additional measure of file security (see section 5.4), more than one set of cryptovariables may be associated with a base register.

6. The IED must be able to switch cryptovariables between memory cycles. This requirement stems from the nature of base register addressing found in modern machines. The number of potential sets of cryptovariables associated with a given job in execution exceeds two and maybe less than 64. If the IED technique is used in connection with file security, many more are required (perhaps several thousands in an extreme case).

7. The IED, in generating cryptovariables must be able to accept an optional, user-supplied keying vairable, that can provide an effective super-encipherment of a particular address space. The keying mechanism must be such that the user-supplied keying variable carries through the transencipherment process, and at a later time permit reading of that information with a different set of short-term cryptovariables only when the user-supplied key is present. This provision permits users to establish some or all of their address space as compartmented, and not readable even by file handling programs or other system programs.

8. Only the operating system (not the user) must be able to change the contents of the base registers and the corresponding cryptovariables. The IED must be designed in such a way that re-assignment of the CPU to another user provides automatic secure safe storage of the cryptovariables associated previous user. Because such re-assignment will be frequent in a time-shared environment the IED should have sufficient storage for all of the cryptovariables for all of the active users* of a system.

---

*Active users: A user whose processes (programs) are known to a system, and are in some state of execution.

5.3.1.3 Secondary Storage Encryption

Objective - To prevent compromise through physical access to memory devices and to provide file isolation.

Method - The secondary storage encryption device is a transencipherment mechanism, connecting directly to externally encrypted files. A number of options for cryptovariables are possible with the device, the simplest being a record address to provide keying variability. The secondary storage encryption device will operate on data in units natural to the secondary storage device.

In order to provide transencipherment, the cryptovariables associated with the users address space containing the data to be read or written must be supplied from the key store, and the device address supplied from the file handling program.

The requirements of the SSED are different from the IED in a number of particulars enumerated below:

1. The SSED is a long term device, with a cryptoperiod measured in months.
2. The SSED must act as a transencipherment device, converting from information in externally encrypted form to internally encrypted form with no intervening plaintext.
3. Keying information for the SSED must be derived from the file(s) identified as part of the users program. This assumes that there is at least one "system" file containing the file directory, from which all other files are identified. It is further assumed that the "system" file is addressable only by certified file handling programs.

4. The SSED must be able to treat information in units "natural" to the specific secondary storage device. For the range of current devices this may bary from 80 to several thousand characters.

5.4 File Security

Objectives - To provide multi-level security to files.

Method

A users requests for data from files is mediated by a file handling component of the operating system. Among its services are the collection of file records and placement of them in a device-sized space (physical record) before writing them onto secondary storage. On input, it accepts physical records from secondary storage, and supplys logical records to a user on demand.

With the services of an internal encryption device, it is possible to provide security to files in a manner that will permit (if desired) a single logical file containing records of various classifications (in the widest sense of the word) to be accessed by authorized users fro only those records to which their clearance permits.

5.4.1 Assumptions for the Internal Encipherment Service for File Security

The Internal Encipherment device (IED) previously described made provision for a user-controlled private and secure section of the memory by utilizing a user-supplied parameter directly as a cryptovariable of the IED. In order to be applied to file security, it is necessary for the IED to have access to as many user-supplied cryptovariables as there are unique "classifications" of the data. Furthermore, it is necessary that the device have "long-term" properties as well as or in lieu of the short-term properties described for the IED previously. Specifically the transform key must have the properties:

$P[T(SK, UK)] \rightarrow C$

$C[T(SK)] \rightarrow C'$

$C'[T(UK)] \rightarrow P$

where [T(SK, UK)] is read "transformed by the cryptologic transformation T, with cryptovariables SK, UK."

It is further assumed that each record of the file contains a header describing the classification and access control category for the record, and that the header may be read by the file handling program (i.e., is enciphered only by a file key or an internal (system key)). The file handling program passes records to a user program encrypted with both an internal key, and the user private key.

The file security capabilities are depicted in the following symbolic steps.

1. D_X $\frac{\wedge|SK}{}$ SB
2. SB $\frac{SK|UK}{}$ UWS
3. UWS $\frac{UK|UKP}{}$ UPDB
4. UPDB $\frac{UK\bar{P}_c|SK}{}$ SB
5. SB $\frac{SK|F_2}{}$ D_X

initial creation of a file.

1. $\frac{F_2|SK'}{}$ SB
2. SB $\frac{SK'|UK'\bar{P}}{}$ UPWS
3. UPWS $\frac{UK'P|UK'P}{}$ UPWS

subsequent referencing of a file

1. Data (D_X) is read from an external device with a null file key (i.e., en claire) into system buffer (SB) storage keyed with a (local) system key.

2. Records are moved to a users work space (UWS), being fetch-keyed with an internal key (SK), and store-keyed with a user key (UK).

3. Reocrd classification is determined by the user, and the record moved from the users work space to a user buffer area that is user-private (UPDB). Fetches are keyed with the user key (UK), and stores keyed with the user key and a user private key appropriate to the classification (UKP).

4. The user buffer area is moved to the system buffer (by the system file handling program). Because the system file handling program cannot have the user ID, fetches are only keyed with the user key portion of the crypto-variables. (UK), while stores are keyed with the internal key. At this point, the record is superenciphered with the users private key.

5. The system buffer storage is written onto an external device via a transencipherment device. The data is fetch-keyed with the internal key for the file handling program (SK) and write-keyed with the file key (F_2).

A subsequent authorized reference to the file operates in a similar manner.

1. The physical records of the file are read into the system buffer. Data is read-keyed with the file key (F_2) and store-keyed with a (possibly new) internal key (SK').

2. The classification data (in plain form) is examined by the file handling programs and those records matching the users access authroization are moved to a user private work space (UPWS). Data is fetch-keyed by an internal key (SK'), and store-keyed by only the user key (UK') portion of the cryptovariable for that user.

3. The user references the data in the record using the private key portion applicable to the data in the records (UK'P).

For files with a large number of individual classifications, it is necessary for the user to have the key(s) corresponding to all of the records to which he is entitled. This implies that the cryptovariable storage be large enough to accomodate all of the user-supplied keys for the various records.

Since the original conception of the IED had a one-to-one correspondence between base registers and cryptovariable storage positions, it may be necessary to "address" cryptovariables because each user-supplied key corresponding to a different record classification would have to be applied to the same users address space. Thus we have a single address space (the record) and multiple cryptovariables (user-supplied) applied to it.

The principle purpose of using encipherment techniques applied to files is to reduce the scope of certification of a system by localizing the points in a system that require certification. Thus, if unauthorized attempts at

access to file records succeeded, nothing is lost since the data is super-enciphered. Users with only limited access to a file cannot read data they are not entitled to even if through error or attack they gain access to the data.

5.5 CERTIFICATION

Objective - To protect against unauthroized recovery of classified information, denial of system usage, and intelligent deception.

Method - Certify that the systems, hardware/software, provides effective countermeasures (e.g., authentication, memory bounds, passwords, etc.) and sufficient alarms in case of their failure or attempted defeats. These countermeasures are those identified by applicable governing standards. Countermeasures required of a multi-level system are similar to those identified for the closed system in order to protect information on the need-to-know basis, though their probability of effectiveness may necessarily be higher. An audit trail would provide a means of assessing possible compromist situations.

Requirements

Software: The entire machine listing must be "wrung-out" to verify that those countermeasures accomplished by the software package are adequate and effective as defined by governing standards and criteria.

Hardware: The components involved in the execution of required countermeasures must be analyzed and the probability of failures resulting in compromise of information must be determined.

System: Adequate operational tests must be performed, both to exercise those countermeasures required and to attest that these cannot be defeated. Operational testing of these countermeasures must be performed with sufficient periodicity to certify that security effectiveness, once established, is maintained.

Conclusion - While reliability studies can be performed on the system hardware to certify its security effectiveness; the software packages associated with multi-level time-shared systems present a formidable task, that, to date, has not been totally accomplished. With regard to the software, not only the security and countermeasures, routines, but the entire listing, and tables, must be certified to assure that these routines are not only sufficient but also appropriately exercised and that there is no threat of their being by-passed. Additionally, any certification must be continual.

A "black box" analyzer/certifier might be developed in order to automate and accelerate the analysis process, but at best such an approach could serve only to certify a specific system.

5.6 Research Recommendations

The conceptual framework for system security outlined above poses a number of questions that cannot be answered from available information. Broadly speaking, the questions are:

1. What is a suitable form for an internal encryption device for time-shared systems? How does the form change as the characteristics of the system vary?
2. What are the cryptologic techniques applicable to internal encryption devices, and transencipherment equipment?
3. To what extent is the problem of software certification reduced by incorporation of encipherment devices in the internal operation of computer systems?
4. Since certification (at some level) is required even with the use of IED's, what technology can be applied to automate or assist in the problem of certifying systems (with or without the use of IED's)?

It is recommended that a research program be initiated to attempt to answer some of these questions. Component of such a program are outlined below.

5.6.1 Structure of Internal Encipherment Devices

The objective of this element of a research program is to develop the systems requirements of internal encipherment and transencipherment devices by describing functionally these devices and how they would operate in a time-shared system. In addition to functional designs of IED's, their cryptologic requirements can be described as well.

5.6.2 Cryptologic Techniques

Specifically, what are suitable cryptologic techniques for use in IED's and transencipherment devices. If the IED's follow the outline presented, it would be necessary for them to be "long term" with respect to user-private keys, and "short term" with respect to system-supplied keys. Furthermore, the requirement that they operate on information in transit between primary storage and the CPU indicates they should be fast enough not to cause any appreciable delay in operation of the system. Furthermore, the IED concept treats programs and data as a large number of identical length messages. Since the content of these "messages" may be assumed in many instances, a wealth of material is available for analysis of the IED. Thus the IED and the other internal security devices may require development and testing of new cryptologic techniques in order to achieve a suitable level of security just as cryptologic devices.

5.6.3 Systems Sturcture and Certification Techniques

This entire area deserves considerable effort on a broad front. Broadly specking, the problem is what are the points of security vulnerability of a system, and what is the behavior of the system under all circumstances surrounding attack on those points? Our present state of capability for specifying systems is either too gross or too detailed to permit much in the way of automated assistence in certifying systems. We need first a rigorous way of specifying both the hardware, and the program of an operating system (at least). This specification must be at some level above the logic equations specifying the system, although it should be possible to descend to the logic level for any or all portions of a

system for detailed analysis. Coupled with rigorous specifications of a system must be methods of expanding the specification, and ways of observing the detailed behavior of parts of a system under different assumptions of programs, data, and configuration. Equally difficult in certification is enumerating unacceptable behavior. A considerable amount of research is required before these problems can be coped with in any reasonable way.